

An Estimation and Simulation Framework for Energy Efficient Design using Platform FPGAs

Sumit Mohanty, Jingzhao Ou, Viktor K. Prasanna
{smohanty, ouj, prasanna}@usc.edu

Abstract

We discuss an estimation and simulation framework for energy efficient application design using platform FPGAs. This framework integrates various widely used simulators available for platform FPGAs, allows specification of kernel designs as a set of libraries, and enables efficient application design through the integration of high-level performance estimators and design space exploration tools. This framework is based on the concept of Model Integrated Computing and provides a flexible, extensible, and plug-n-play environment for application design.

1. Background and Motivation

Platform FPGAs integrate (within the FPGA fabric) processors for system control and DSP cores for customized data filtering and parallel processing [8]. These devices offer several design choices such as operating frequency, precision, amount of memory, degree of parallelism, clock gating, implementation alternatives, etc. These choices define a large design space that must be efficiently explored to find suitable solutions. Adaptive beam forming, multi-rate filters and wavelets, software defined radio, etc. are some of the target applications for platform FPGAs. These applications are compute intensive and multi-functional, are expected to satisfy hard latency constraints, and many of them are candidates for deployment in power constrained environments. Thus, in addition to latency and area, energy is a key performance metric. Our earlier work has focused on a high-level modeling technique, domain specific modeling, and a methodology for energy-efficient design of application kernels using FPGAs [1, 5]. In this paper, we discuss an estimation and simulation framework for energy and latency estimation for application designs using platform FPGAs.

The framework discussed in this paper is an extension of the MILAN design environment [3]. MILAN is a Model based Integrated simuLAtioN framework for embedded system design and optimization through integration of various simulators into a unified environment. Using the MILAN framework, a designer formally models the target application and architecture. The information captured in the

models are used to drive various simulators from a single interface. MILAN is an extensible framework where the modeling capability can be extended to support additional domains and integrate additional simulators. The extension of MILAN, discussed in this paper, modifies the modeling paradigm to support modeling of designs using Platform FPGAs, develops a performance estimator for kernel designs, and integrates simulators specific to the Platform FPGAs. We also enhance HiPerE, a high-level, system-wide, performance estimator to support Platform FPGAs [6].

The framework discussed in this paper adopts Model Integrated Computing (MIC) as the core design philosophy [2]. MIC advocates the extension of the scope and use of models so that they form the backbone of system development. For a given domain (e.g. platform FPGAs), using integrated multiple-view models to capture all aspects (application, resource, behavior, constraints, etc.) of a system it is possible to drive various tools and simulators using a single model. The Generic Modeling Environment (GME) is a configurable graphical tool suite supporting MIC [2]. GME is configured through the use of metamodels which contain syntactic, semantic, and presentation specifications of the target domain [2]. Model Interpreters (MI) are tool specific software components that translate the information captured in the models into the format required by the tools enabling tool-integration.

2. Estimation and Simulation Framework

The framework consists of three modules, integrated simulation module, kernel specification module, and system-level design module (Figure 1).

A. Integrated Simulation Module

This module integrates a set of widely used power and latency simulators available for platform FPGAs. The inputs to this module are specifications in HDL or C and appropriate input stimulus (both generated from the models) and the outputs are various performance estimates. The HDL or C specification is generated based on the model of the hardware and application respectively [3]. Currently, the framework targets the Xilinx Virtex-II Pro [8]. The specific simulators and related tools that are (being) integrated into the

framework are ModelSim and Power Estimator, XPower, XST, and Place and Route tools from Xilinx [4, 8]. For PowerPC simulation, we are currently using ModelSim and GDB Insight. All the tools are driven from a single modeling paradigm. Additional simulators can be easily integrated by writing appropriate MIs which makes the framework plug-n-play. To improve accuracy, we simulate each design a number of times with randomly generated input waveforms with varying levels of switching activity and evaluate the average latency and energy using a confidence interval for statistically significant results. We have also integrated two performance estimators, Kernel Performance Estimator and HiPerE, that are discussed in the following sections.

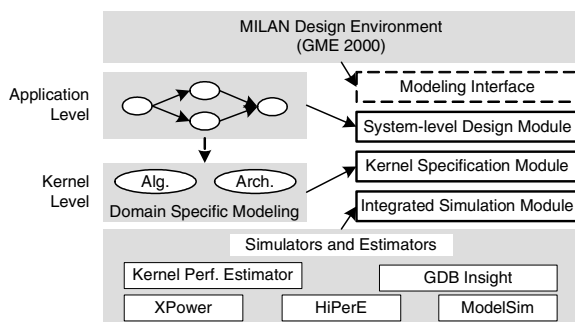


Figure 1. Estimation and Simulation Framework

B. Kernel Specification Module

The kernel specification module provides the required modeling support to capture a parameterized kernel design. This module is developed based on the domain specific modeling technique discussed in [1]. Given a design for a specific kernel, it is modeled using micro modules such as registers, multipliers, adders, multiplexers, interconnect lines, etc. and macro modules such as processing elements, block of memory, etc. as building blocks. All building blocks are parameterized and are pre-characterized for area and power consumption. The pre-characterization step involves simulation of each building block at different design points (various combination of parameter values). The simulation result is provided as feedback to the framework to characterize power dissipation and area as functions of the parameters. Switching activity is a default parameter for power dissipation for the building blocks. We have developed a tool, Kernel Performance Estimator, which given a datapath design using these basic building blocks and algorithm details in terms of cycles required and activity for each building block, provides rapid performance and area estimates [1]. The kernel specifications are also stored as a library of designs which essentially consists of a set of models. These models are later used for complete application design and the kernel performance estimator provides the performance estimates.

C. System-level Design Module

This module provides various support for system-level design. The designer initially models the application as a set of tasks provided as a DAG and identifies appropriate kernel designs from the library available in the kernel specification module for each task. A large number of choices for individual kernels results in a large design space. DESERT, a design space exploration (DSE) tool integrated into MILAN, is used to identify a set of designs that satisfy user specified design constraints [7]. HiPerE is used to rapidly estimate various performance metrics and perform a more focused DSE following DESERT [6]. HiPerE is designed to consider parallelism, reconfiguration, data transfer cost, memory access cost, etc. while estimating performance of a design and therefore can provide reasonably accurate estimates. HiPerE also generates an activity report that provides a coarse profile of a design that summarizes activity of the processors, memory in terms of task execution, idle-time, voltage scaling, reconfiguration, etc. Finally, the low-level simulators are used to simulate a reduced number (typically 10 – 20) of designs to identify the final design.

3. Conclusion

We presented an estimation and simulation framework for energy-efficient application design using FPGAs. Choice of simulators enables hierarchical simulation which provides the designer the ability to select faster (possibly less accurate) simulators initially and use slower and accurate simulators in the later phases of application design exploiting the speed versus accuracy trade-off among the simulators. Ability to store kernel designs as library of models promotes design reuse through the sharing of the kernel designs among applications.

Acknowledgment: This work is part of the PARIS project, supported by the DARPA PAC/C program under contract S7-6AX077X-62A9.

References

- [1] S. Choi, J.-W. Jang, S. Mohanty, and V. K. Prasanna. Domain-specific modeling for rapid system-wide energy estimation of reconfigurable architectures. In *Engineering of Reconfigurable Systems and Algorithms*, 2002.
- [2] Generic Modeling Environment. <http://www.isis.vanderbilt.edu>.
- [3] Model-based Integrated Simulation. <http://milan.usc.edu/>.
- [4] Modelsim, Model Technologies. <http://www.model.com/>.
- [5] S. Mohanty, S. Choi, J. Jang, and V. K. Prasanna. A model-based methodology for application specific energy efficient data path design using fpgas. In *Conference on Application-specific Systems, Architectures and Processors*, 2002.
- [6] S. Mohanty and V. K. Prasanna. Rapid system-wide performance evaluation and optimization for heterogeneous soc architectures. In *IEEE ASIC/SOC Conference*, 2002.
- [7] S. Neema. *System Level Synthesis of Adaptive Computing Systems*. PhD thesis, Dept. of Electrical and Computer Engg., Vanderbilt University, 2001.
- [8] Xilinx application note: Virtex-II series and Xilinx ISE 4.1i design environment. <http://www.xilinx.com/>.